PATENT ABSTRACTS OF JAPAN

(11)Publication number : 2002-268659
(43)Date of publication of application : 20.09.2002

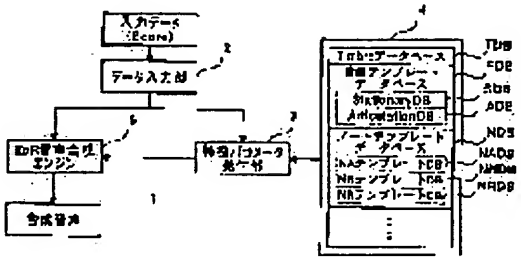
(51)Int.Cl. G10L 13/00
G06F 3/16
G10L 13/08
G10L 13/06

(21)Application number : 2001-067258
(22)Date of filing : 09.03.2001

(71)Applicant : YAMAHA CORP
(72)Inventor : HISAMINATO YUJI
JORDI BONADA

(54) VOICE SYNTHESIZING DEVICE

(57)Abstract:
PROBLEM TO BE SOLVED: To provide a voice synthesizing device which reduces the size of a database while minimizing deterioration in voice quality.
SOLUTION: The voice synthesizing device has a storage means which stores phoneme pieces having different pitches by phonemes represented as the same phoneme symbols, a readout means which reads the phoneme pieces out by using the pitches as indexes, and a voice synthesizing means which synthesizes a voice according to the read phoneme pieces.



LEGAL STATUS

[Date of request for examination] 19.06.2003

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

(19)日本国特許庁（ＪＰ）　(12)公開特許公報（Ａ）　(11)特許出願公開番号
特開2002－268659
（P2002－268659A）
(43)公開日　平成14年９月20日(2002.9.20)

(51)Int.Cl.⁷	識別記号	ＦＩ		テーマコード（参考）
Ｇ１０Ｌ　13/00		Ｇ０６Ｆ　3/16		３３０Ｋ　５Ｄ０４５
Ｇ０６Ｆ　3/16	３３０	Ｇ１０Ｌ　3/00		Ｊ
Ｇ１０Ｌ　13/08				Ｈ
13/06			5/04	Ｆ
			9/02	Ｌ

審査請求　未請求　請求項の数13　ＯＬ　（全 20 頁）

(21)出願番号　特願2001－67258(P2001－67258)

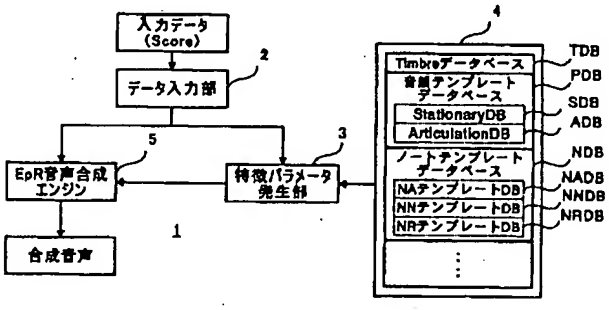
(22)出願日　平成13年３月９日(2001.3.9)

(71)出願人　000004075
ヤマハ株式会社
静岡県浜松市中沢町10番１号
(72)発明者　久湊　裕司
静岡県浜松市中沢町10番１号　ヤマハ株式会社内
(72)発明者　ジョルディ　ボナダ
パッセィグデェ・シィルクインバラシィオ・８．08003　バルセロナ　スペイン
(74)代理人　100091340
弁理士　高橋　敬四郎　（外２名）

Ｆターム(参考)　5D045 AA07 AA20

(54)【発明の名称】　音声合成装置

(57)【要約】
【課題】　音質の劣化を最小限に抑えつつ、データベースのサイズを縮小した音声合成装置を提供する。
【解決手段】　音声合成装置は、同じ音素記号で表される音素毎に複数の異なるピッチを持つ音素片を記憶する記憶手段と、前記音素片をピッチをインデックスとして読み出す読出し手段と、前記読み出された音素片に基き音声を合成する音声合成手段とを有する。



【特許請求の範囲】

【請求項１】　同じ音素記号で表される音素毎に複数の異なるピッチを持つ音素片を記憶する記憶手段と、
前記音素片をピッチをインデックスとして読み出す読出し手段と、
前記読み出された音素片に基き音声を合成する音声合成手段とを有する音声合成装置。

【請求項２】　同じ音素記号で表される音素毎に複数の異なる音楽表現度を持つ音素片を記憶する記憶手段と、
前記音素片を音楽表現度をインデックスとして読み出す読出し手段と、
前記読み出された音素片に基き音声を合成する音声合成手段とを有する音声合成装置。

【請求項３】　同じ音素記号で表される音素毎に複数の異なる音素片を記憶する記憶手段と、
音声合成の為の音声情報を入力する入力手段と、
前記音声情報に合致する音素片が前記記憶手段に記憶されていない場合に、前記記憶手段に記憶されている音素片を元に前記音声情報に合致する音素片を補間して算出する補間手段と、
前記補間して算出された音素片に基づき音声を合成する音声合成手段とを有する音声合成装置。

【請求項４】　音声の特徴量の時間変化分をテンプレートデータとして記憶する記憶手段と、
音声合成の為の音声情報を入力する入力手段と、
前記音声情報に基づき前記テンプレートデータを前記記憶手段から読み出す読出し手段と、
前記読み出されたテンプレートデータと、前記音声情報に基づき音声を合成する音声合成手段とを有する音声合成装置。

【請求項５】　同じ音素記号で表される音素毎に複数の異なるピッチを持つ音素片を記憶する記憶手段から、ピッチをインデックスとして音素片を読み出す読出し工程と前記読み出された音素片に基き音声を合成する音声合成工程とを有する音声合成方法。

【請求項６】　同じ音素記号で表される音素毎に複数の異なる音楽表現度を持つ音素片を記憶する記憶手段から、音楽表現度をインデックスとして音素片を読み出す読出し工程と前記読み出された音素片に基き音声を合成する音声合成工程とを有する音声合成方法。

【請求項７】　同じ音素記号で表される音素毎に複数の異なる音素片を記憶する記憶手段から音素片を読み出す読出し工程と、
音声合成の為の音声情報を入力する入力工程と、
前記音声情報に合致する音素片が前記記憶手段に記憶されていない場合に、前記記憶手段に記憶されている音素片を元に前記音声情報に合致する音素片を補間して算出する補間工程と、
前記補間して算出された音素片に基づき音声を合成する音声合成工程とを有する音声合成方法。

【請求項８】　音声の特徴量の時間変化分をテンプレートデータとして記憶する記憶工程と、
音声合成の為の音声情報を入力する入力工程と、
前記音声情報に基づき音声の特徴量の時間変化分を表すテンプレートデータを前記記憶手段から読み出す読出し工程と、
前記読み出されたテンプレートデータと、前記音声情報に基づき音声を合成する音声合成工程とを有する音声合成方法。

【請求項９】　同じ音素記号で表される音素毎に複数の異なるピッチを持つ音素片を記憶する記憶手段から、ピッチをインデックスとして音素片を読み出す読出し手順と前記読み出された音素片に基き音声を合成する音声合成手順とを有する音声合成手順をコンピュータに実行させるためのプログラム。

【請求項１０】　同じ音素記号で表される音素毎に複数の異なる音楽表現度を持つ音素片を記憶する記憶手段から、音楽表現度をインデックスとして音素片を読み出す読出し手順と前記読み出された音素片に基き音声を合成する音声合成手順とを有する音声合成手順をコンピュータに実行させるためのプログラム。

【請求項１１】　同じ音素記号で表される音素毎に複数の異なる音素片を記憶する記憶手段から音素片を読み出す読出し手順と、
音声合成の為の音声情報を入力する入力手順と、
前記音声情報に合致する音素片が前記記憶手段に記憶されていない場合に、前記記憶手順に記憶されている音素片を元に前記音声情報に合致する音素片を補間して算出する補間手順と、
前記補間して算出された音素片に基づき音声を合成する音声合成手順とを有する音声合成手順をコンピュータに実行させるためのプログラム。

【請求項１２】　音声の特徴量の時間変化分をテンプレートデータとして記憶する記憶手順と、
音声合成の為の音声情報を入力する入力手順と、
前記音声情報に基づき音声の特徴量の時間変化分を表すテンプレートデータを記憶手段から読み出す読出し手順と、
前記読み出されたテンプレートデータと、前記音声情報に基づき音声を合成する音声合成手順とを有する音声合成手順をコンピュータに実行させるためのプログラム。

【請求項１３】　同じ音素記号で表される音素毎に複数の異なるピッチを持つ音素片と、
音声の特徴量の時間変化分を表すテンプレートデータであって、前記音素片に対して適用されるテンプレートデータとを有する音声合成用データベースを記憶した媒体。

【発明の詳細な説明】

【０００１】

【発明の属する技術分野】本発明は、音声合成装置に関

し、より詳しくは、人間の歌唱音声を合成する音声合成装置に関する。

【０００２】

【従来の技術】人間の音声は、音韻（音素）により構成され、各音韻は複数個のフォルマントにより構成されている。よって、人間の歌唱音声の合成は、まず、人間が発生することのできる全ての音韻に対して、その各音韻を構成する全てのフォルマントを発生して合成できるように準備し、必要な音韻を生成する。次に、生成された複数の音韻を順次つなぎ合わせ、メロディに合わせて音高を制御する。この手法は、人間の音声に限らず、フォルマントを有する楽音、例えば、管楽器から発生される楽音の合成にも適用できる。

【０００３】この手法を用いた音声合成装置は従来から知られており、例えば、特許公報第２５０４１７２号には、高い音高のフォルマント音を発生するときでも、不要なスペクトルを発生しないように構成したフォルマント音発生装置が開示されている。

【０００４】また、フォルマント周波数は、ピッチに依存することが知られており、特開平６－３０８９９７号公報の実施例に記載されているように、ピッチ周波数ごとにいくつかの音素片をデータベースに持っておき、音声のピッチに従って、適切な音素片を選択する技術が知られている。

【０００５】

【発明が解決しようとする課題】しかし、上記のような従来のデータベースでは、１つの音素片について、一定以上数のピッチ周波数の音素片を持つ必要があり、データベースのサイズが、比較的大きくなってしまう。

【０００６】また、多くの異なるピッチで発生された音声から音素片を抽出する必要があるために、データベースの構築に時間を要する。

【０００７】さらには、フォルマント周波数は、ピッチのみに依存するのではなく、他の要素、例えば、ダイナミクス等が加わることにより、二乗、三乗とデータ量が増えてしまう。

【０００８】本発明の目的は、音質の劣化を最小限に抑えつつ、データベースのサイズを縮小した音声合成装置を提供することである。

【０００９】また、本発明の他の目的は、上記データベースを用いた音声合成装置を提供することである。

【００１０】

【課題を解決するための手段】本発明の一観点によれば、音声合成装置は、同じ音素記号で表される音素毎に複数の異なるピッチを持つ音素片を記憶する記憶手段と、前記音素片をピッチをインデックスとして読み出す読出し手段と、前記読み出された音素片に基き音声を合成する音声合成手段とを有する。

【００１１】また、本発明の他の観点によれば、音声合成装置は、同じ音素記号で表される音素毎に複数の異なる音楽表現度を持つ音素片を記憶する記憶手段と、前記音素片を音楽表現度をインデックスとして読み出す読出し手段と、前記読み出された音素片に基き音声を合成する音声合成手段とを有する。

【００１２】また、本発明の他の観点によれば、音声合成装置は、同じ音素記号で表される音素毎に複数の異なる音素片を記憶する記憶手段と、音声合成の為の音声情報を入力する入力手段と、前記音声情報に合致する音素片が前記記憶手段に記憶されていない場合に、前記記憶手段に記憶されている音素片を元に前記音声情報に合致する音素片を補間して算出する補間手段と、前記補間して算出された音素片に基づき音声を合成する音声合成手段とを有する。

【００１３】また、本発明の他の観点によれば、音声合成装置は、音声の特徴量の時間変化分をテンプレートデータとして記憶する記憶手段と、音声合成の為の音声情報を入力する入力手段と、前記音声情報に基づき前記テンプレートデータを前記記憶手段から読み出す読出し手段と、前記読み出されたテンプレートデータと、前記音声情報に基づき音声を合成する音声合成手段とを有する。

【００１４】

【発明の実施の形態】図１は、音声合成装置１の構成を表すブロック図である。

【００１５】音声合成装置１は、データ入力部２、特徴パラメータ発生部３、データベース４、ＥｐＲ音声合成エンジン５を有する。

【００１６】データ入力部２に入力される入力データＳｃｏｒｅは、特徴パラメータ発生部３及びＥｐＲ音声合成エンジン５に送られる。特徴パラメータ発生部３は、入力データＳｃｏｒｅに基づきデータベース４から後述する特徴パラメータ、各種テンプレートを読み込む。特徴パラメータ発生部３は、さらに、読み込んだ特徴パラメータに各種テンプレートを適用して、最終的な特徴パラメータを生成してＥｐＲ音声合成エンジン５に送る。

【００１７】ＥｐＲ音声合成エンジン５では、入力データＳｃｏｒｅのピッチ、ダイナミクス等に基づきパルスを発生させ、該発生させたパルスに特徴パラメータを適用することにより、音声を合成して出力する。

【００１８】図２は、入力データＳｃｏｒｅの一例を示す概念図である。音韻トラックＰＨＴ、ノートトラックＮＴ、ピッチトラックＰＩＴ、ダイナミクストラックＤＹＴ、オープニングトラックＯＴによって構成されており、楽曲のフレーズ若しくは曲全体の、時間とともに変化するデータが保存されている楽曲データである。

【００１９】音韻トラックＰＨＴには、音韻名と、その発音継続時間が含まれる。さらに、各音韻は、音素と音素の遷移部分であることを示すアーティキュレーション（Ａｒｔｉｃｕｌａｔｉｏｎ）とその他の定常部分であることを示すステーショナリー（Ｓｔａｔｉｏｎａｒ

ｙ）との２つに分類される。各音韻は、これらのうちどちらに分類されるかに付いてのフラグも含むものとする。なお、アーティキュレーションは、遷移部分であるので、先頭音韻名と後続音韻名の複数の音韻名を有している。一方、ステーショナリーは定常部分であるので１つの音韻名だけからなる。

【００２０】ノートトラックＮＴには、ノートアタック（ＮｏｔｅＡｔｔａｃｋ）、ノートトゥノート（ＮｏｔｅＴｏＮｏｔｅ）、ノートリリース（ＮｏｔｅＲｅｌｅａｓｅ）のいずれかを示すフラグが記録されている。ノートアタックは発音の立ち上がり時、ノートトゥノートは音程の変化時、ノートリリースは発音の立下り時の音楽表現を指示するコマンドである。

【００２１】ピッチトラックＰＩＴには、各時刻において発音すべき音声の基本周波数が記録されている。なお、実際に発音される音声のピッチはこのピッチトラックＰＩＴに記録されているピッチ情報に基づき他の情報を用いて算出されるので、実際に発音されているピッチと、ここに記録されているピッチは異なる場合がある。

【００２２】ダイナミクストラックＤＹＴには、音声の強さを示すパラメータである各時刻におけるダイナミクス値が記録されている。ダイナミクス値は、０から１までの値をとる。

【００２３】オープニングトラックＯＴには、唇の開き具合（唇開度）を示すパラメータである各時刻のオープニング値が記録されている。オープニング値は０から１までの値をとる。

【００２４】特徴パラメータ発生部３は、データ入力部２から入力される入力データＳｃｏｒｅに基づき、データベース４からデータを読み出し、後述するように、入力データＳｃｏｒｅ及びデータベース４から読み出したデータに基づき特徴パラメータを発生して、ＥｐＲ音声合成エンジン５に出力する。

【００２５】この特徴パラメータ発生部３で発生する特徴パラメータは、例えば、励起波形スペクトルのエンベロープ、励起レゾナンス、フォルマント、差分スペクトルの４つに分類することが出来る。これらの４つの特徴パラメータは、実際の人間の音声等（オリジナルの音声）を分析して得られる調和成分のスペクトル・エンベロープ（オリジナルのスペクトル）を分解することにより得られるものである。

【００２６】励起波形スペクトルのエンベロープ（ＥｘｃｉｔａｔｉｏｎＣｕｒｖｅ）は、声帯波形の大きさ（ｄＢ）を表すＥＧａｉｎ、声帯波形のスペクトルエンベロープの傾きを表すＥＳｌｏｐｅＤｅｐｔｈ、声帯波形のスペクトルエンベロープの最大値から最小値の深さ（ｄＢ）を表すＥＳｌｏｐｅの３つのパラメータによって構成されており、以下の式（Ａ）で表すことが出来る。

【００２７】

【数式１】

$$ExcitationCurve(f) = EGain + ESlopeDepth * (\exp(-ESlope * f) - 1) \quad \cdots (A)$$

励起レゾナンスは、胸部による共鳴を表す。中心周波数（ＥＲＦｒｅｑ）、バンド幅（ＥＲＢＷ）、アンプリチュード（ＥＲＡｍｐ）の３つのパラメータで構成され、２次フィルター特性を有している。

【００２８】フォルマントは、１から１２個のレゾナンスを組み合わせることにより声道による共鳴を表す。中心周波数（ＦｏｒｍａｎｔＦｒｅｑ$_i$）、バンド幅（ＦｏｒｍａｎｔＢＷ$_i$）、アンプリチュード（ＦｏｒｍａｎｔＡｍｐ$_i$）の３つのパラメータで構成される。なお、「ｉ」は、１から１２までの値（$1 \leqq i \leqq 12$）である。

【００２９】差分スペクトルは、上記の励起波形スペクトルのエンベロープ、励起レゾナンス、フォルマントの３つで表現することの出来ないオリジナルスペクトルとの差分のスペクトルを持つ特徴パラメータである。

【００３０】データベース４は、少なくともＴｉｍｂｒｅデータベースＴＤＢ、音韻テンプレートデータベースＰＤＢ、ノートテンプレートデータベースＮＤＢから構成されている。

【００３１】一般に、ＴｉｍｂｒｅデータベースＴＤＢに保存されている特定の時刻から得られた特徴パラメータのみを用いて音声を合成した場合には非常に単調で、機械的な音声になる。また、音素が連続する場合にはその遷移部分での音声は実際には徐々に変化してゆくので、音素の定常部分のみを単純に連結した場合には、接続点では非常に不自然な音声となる。そこで音韻テンプレート、及びノートテンプレートをデータベースとして持ち、音声合成時に使用することにより、それらの欠点を低減することが可能となる。

【００３２】Ｔｉｍｂｅｒとは音韻の音色であり、ある時刻１点における特徴パラメータ（励起スペクトル、励起レゾナンス、フォルマント、差分スペクトルのセット）で表現される。図３にＴｉｍｂｒｅデータベースＴＤＢの例を示す。このデータベースは、インデックスとして音韻名、ピッチを持つ。

【００３３】なお、以下、この明細書では図３に示すＴｉｍｂｒｅデータベースＴＤＢを使うが、より細かく特徴パラメータを指定できるように、図４に示すようにインデックスとして音韻名、ピッチ、ダイナミクス、オープニングの４つを持つデータベースを用意してもよい。

【００３４】音韻テンプレートデータベースＰＤＢはステーショナリーテンプレートデータベースとアーティキュレーションテンプレートデータベースで構成される。ここでテンプレートとは、特徴パラメータＰとピッチＰｉｔｃｈのペアが一定時間ごとに並んだシーケンス、及

び、その区間の長さＴ（ｓｅｃ．）の組であり、以下の式（Ｂ）で表すことが出来る。

【００３５】

【数式２】

$$Template = \{P(t), Pitch(t), T\}$$

…（Ｂ）

なお、ｔ＝０、Δｔ、２Δｔ、３Δｔ、…、Ｔであり、本実施例では、Δｔは５ｍｓとする。

【００３６】Δｔを小さくすると時間分解能がよくなるので音質は良くなるがデータベースのサイズが大きくなり、逆にΔｔを大きくすると音質が悪くなるがデータベースのサイズは小さくなる。Δｔを決定する際には音質とデータベースのサイズとの優先度を考慮して決定すればよい。

【００３７】図５は、ステーショナリーテンプレートデータベースの一例である。ステーショナリーテンプレートデータベースは、音韻名と代表ピッチをインデックスとして、すべての有声の音韻についてのステーショナリーテンプレートを有している。ステーショナリーテンプレートは音韻、ピッチの安定した部分の音声をＥｐＲモデルを使って分析することによって得ることができる。

【００３８】あるひとつの有声音、例えば「あ」、を長く伸ばして、ある音程、例えばＣ４、で発声した場合にはピッチやフォルマント周波数などの特徴パラメータは、ほぼ一定であり定常（ステーショナリー）であると言えるが、実際には若干の変動が生じている。この変動がなく完全に一定の場合には無機質で機械的な音声になってしまい、逆に言えば、その変動が人間らしさ、自然性を表すと言える。

【００３９】有声音を合成する場合に、Ｔｉｍｂｒｅ、つまりある時刻１点の特徴パラメータのみを使うのではなく、それにステーショナリーテンプレートにある実際の人間の音声から取り出した特徴パラメータの時間変動分、ピッチ変動分を加算することによって有声音に自然性を与えることができる。

【００４０】歌唱音声合成の場合には音符の長さに従って発音する時間を変化させる必要があるが、十分長いテンプレートを１つだけ用意する。テンプレートよりも長い有声音を合成する場合には、テンプレートの時間軸の伸縮をすることはしないで、テンプレートの持っている時間をそのままにして有声音の先頭部分からテンプレートを適用する。

【００４１】テンプレートの終端まで達したら、その後に再び同じテンプレートを繰り返し適用する。なお、テンプレートの終端まで達したら、テンプレートの時間を逆にしたテンプレートを適用する方法も考えられる。この方法ではテンプレートの接続点での不連続がなくなる。

【００４２】テンプレートの時間軸を伸縮することをしないのは、特徴パラメータ、ピッチの変動のスピードが大きく変わると自然性が損なわれるからである。定常部分の揺らぎは人間が意識してコントロールするものではないという考え方からも伸縮しない方が好ましい。

【００４３】ステーショナリーテンプレートは、定常部分の特徴パラメータの時系列をそのまま持つのではなく、その音素の代表的な特徴パラメータと、その変動量を持つ構造である、定常部分の特徴パラメータの変動量は小さいことから、特徴パラメータをそのまま持つことに比べて、変動量で持つ方が情報量が少なく、データベースのサイズを小さくする効果がある。

【００４４】図６はアーティキュレーションテンプレートデータベースの一例である。アーティキュレーションテンプレートデータベースは、先頭音韻名と後続音韻名と代表ピッチとをインデックスとしている。アーティキュレーションテンプレートデータベースには、一定の言語における現実的に可能な音韻の組合せについてアーティキュレーションテンプレートが保存されている。

【００４５】アーティキュレーションテンプレートはピッチの安定した、音韻の接続部分の音声をＥｐＲモデルを使って分析することによって得ることができる。

【００４６】なお、特徴パラメータＰ（ｔ）は絶対値そのままでもいいが、差分値を用いることも出来る。後述するように、合成時には、これらのテンプレートの値の絶対値がそのまま利用されるのではなく、パラメータの相対的な変化量が利用されるので、テンプレートの適用方法に従って、以下の式（Ｃ１）～（Ｃ３）に示すようにＰ（ｔ＝Ｔ）からの差分、あるいはＰ（０）からの差分、あるいはＰ（０）とＰ（Ｔ）を直線で結んだ値との差分の形で特徴パラメータを記録する。

【００４７】

【数式３】

$$Template1 = \{P(t) - P(T), Pitch(t) - Pitch(T), T\}$$

…（Ｃ１）

【数式４】

$$Template2 = \{P(t) - P(0), Pitch(t) - Pitch(0), T\}$$

…（Ｃ２）

【数式５】

$$Template3 = \begin{Bmatrix} P(t) - ((P(T) - P(0)) * t / T + P(0)), \\ Pitch(t) - ((Pitch(T) - Pitch(0)) * t / T + Pitch(0)), T \end{Bmatrix}$$

…（Ｃ３）

人間が２つの音素を連続して発音する場合には、突然変化するのではなくゆるやかに移行していくので、例えば、「あ」という母音の後に区切りを置かないで連続し

て「え」という母音を発音する場合には、最初に「あ」が発音され「あ」と「え」の中間に位置する発音を経て「え」に変化する。

【００４８】この現象は一般に調音結合と呼ばれる現象である。音素の結合部分が自然になるように音声合成を行うには、ある言語において組合せ可能な音素の組合せについて、結合部分の音声情報を何らかの形で持つことが好ましい。

【００４９】音素の結合部分をＬＰＣ係数や音声波形といった形でそのまま持つ方式はすでに存在しているが、本実施例では、特徴パラメータ、ピッチの差分情報を持ったアーティキュレーションテンプレートを使って２つの音素間の調音（Ａｒｔｉｃｕｌａｔｉｏｎ）部分を合成している。

【００５０】例えば、２つの連続する同じ音程の４分音符で、それぞれの歌詞が「あ」、「い」という歌唱を合成する場合を考える。２つの音符の境界には「あ」から「い」への移行部分が存在する。「あ」、「い」は両方とも母音であり、有声音であるので、Ｖ（有声音）からＶ（有声音）へのアーティキュレーションに該当し、後述するタイプ３の方法でアーティキュレーションテンプレートを適用して移行部分の特徴パラメータを求めることができる。

【００５１】すなわち、「あ」と「い」の特徴パラメータをＴｉｍｂｒｅデータベースＴＤＢから読み出し、それらに「あ」から「い」へのアーティキュレーションテンプレートを適用すれば、その移行部分の、自然な変化を持つ特徴パラメータが得られる。

【００５２】ここで、「あ」から「い」への移行部分の時間を、その部分に適用するアーティキュレーションテンプレートの元々の時間と同じにすれば、テンプレートを作成するときに利用した音声波形と同じ変化を得る事が出来る。

【００５３】テンプレートの時間よりもゆっくりと、あるいは長く変化する音声を合成する場合には、テンプレートの長さを線形に伸長してから特徴パラメータの差分を加算すればよい。先に説明したステーショナリーと異なり、２つの音素間の変化部分のスピードは意識的にコントロールできるものであるため、線形にテンプレートを伸縮しても大きな不自然性は生じない。

【００５４】次に２つの連続する同じ音程の４分音符で、それぞれの歌詞が「あ」、「す」という歌唱を合成する場合を考える。２つの音符の境界には「あ」から「す」の子音部分への短い移行部分が存在する。これはＶ（有声音）からＵ（無声音）へのアーティキュレーションに該当するので、後述するタイプ１の方法でアーティキュレーションテンプレートを適用することで移行部分の特徴パラメータを求めることができる。

【００５５】「あ」の特徴パラメータをＴｉｍｂｒｅデータベースＴＤＢより求めて、それに「ａ」から「ｓ」へのアーティキュレーションテンプレートを適用することで、自然な変化を持つ移行部分の特徴パラメータを得る事が出来る。

【００５６】Ｖ（有声音）からＵ（無声音）へのアーティキュレーションで、タイプ１、つまりテンプレートの先頭部分からの差分、を使う理由は、単純に終端部分にあたるＵ（無声音）部分にはピッチ、特徴パラメータが存在しないためである。

【００５７】「す」はローマ字であらわすと「ｓｕ」であり、子音部分「ｓ」と母音部分「ｕ」から構成される。この中間点にも、「ｓ」の音を残しながら「ｕ」が発音される移行部分が存在する。これはＵからＶへのアーティキュレーションに該当するので、ここでもまたタイプ１の方法でアーティキュレーションテンプレートを適用する。

【００５８】「う（ｕ）」の特徴パラメータをＴｉｍｂｒｅデータベースＴＤＢから読み出し、それに「ｓ」から「ｕ」へのアーティキュレーションテンプレートを適用することで、「ｓ」から「ｕ」への変化部分の特徴パラメータを得ることができる。

【００５９】特徴パラメータの差分情報を持ったアーティキュレーションテンプレートは、絶対値で特徴パラメータを記録したテンプレートに比べて、データサイズが少なくなるという利点を持っている。

【００６０】ノートテンプレートデータベースＮＤＢは、少なくとも、ノートアタックテンプレート（ＮＡテンプレート）データベースＮＡＤＢ、ノートリリーステンプレート（ＮＲテンプレート）データベースＮＲＤＢ、ノートトゥノートテンプレート（ＮＮテンプレート）データベースＮＮＤＢを含んでいる。

【００６１】図７はＮＡテンプレートデータベースＮＡＤＢの一例である。ＮＡテンプレートには音声の立ち上がり部分の特徴パラメータ及びピッチの変化情報が含まれている。

【００６２】ＮＡテンプレートデータベースＮＡＤＢには、音韻名と代表ピッチをインデックスとして、すべての有声の音韻についてのＮＡテンプレートが保存されている。ＮＡテンプレートは、実際に発音した音声の立ち上がり部分を分析することによって得られる。

【００６３】ＮＲテンプレートには音声の立下り部分の特徴パラメータ及びピッチの変化情報が含まれている。ＮＲテンプレートデータベースＮＲＤＢはＮＡテンプレートデータベースＮＡＤＢと同じ構造であり、音韻名と代表ピッチをインデックスとして、すべての有声の音韻についてのＮＲテンプレートを持っている。

【００６４】一定のピッチである音素、例えば「あ」を発声しようとしたときの立ち上がり部分（Ａｔｔａｃｋ）を分析すると振幅が徐々に大きくなり、一定のレベルになって安定していくことがわかる。振幅値だけではなく、フォルマント周波数、フォルマントバンド幅、ピ

ッチについても変化している。

【００６５】人間の実際に発声した音声、例えば「あ」、の立ち上がり部分を解析して得たＮＡテンプレートを、定常部分の特徴パラメータに通用することで、その立ち上がり部分の人の音声の持つ自然な変化を与えることができる。

【００６６】すべての音素ごとにＮＡテンプレートを用意すれば、どの音素についてもアタック部分の変化を与えることが可能になる。

【００６７】歌唱では、音楽的に表情をつけるために立ち上がりを速くしたり、ゆったりと歌う場合がある。ＮＡテンプレートは、あるひとつの立ち上がりの時間を持っているが、もともとＮＡテンプレートの持っている速さよりも速く、若しくは遅くすることは、テンプレートの時間軸を線形に伸縮してから適用することで可能になる。

【００６８】テンプレートを伸縮しても、数倍の範囲内ならば、アタックに不自然さは生じないことが実験によりわかっている。より広範囲のアタックの長さを指定して合成できるようにするには、数段階の長さのＮＡテンプレートを用意して、最も長さの近いテンプレートを選択して伸縮するなどの方法を使う。

【００６９】発声の終了する部分、つまり立下り（Ｒｅｌｅａｓｅ）についても、立ち上がり（Ａｔｔａｃｋ）と同様に振幅、ピッチ、フォルマントが変化する。

【００７０】立下り部分に人間の音声の持つ自然な変化を与えるのは、人間が実際に発声した音声の立ち下がり部分を解析して得たＮＲテンプレートを、立下りの開始する前の音素の特徴パラメータに対して適用することで可能となる。

【００７１】図８は、ＮＮテンプレートデータベースＮＮＤＢの一例である。ＮＮテンプレートはピッチが変化する部分の音声の特徴パラメータを持っている。ＮＮテンプレートデータベースＮＮＤＢには、音韻名、テンプレートの始点時刻のピッチ、終了時刻のピッチをインデックスとして、すべての有声の音韻についてのＮＮテンプレートが保存されている。

【００７２】ピッチの異なる２つの音符を連続して間を置かずに歌唱するときに、前の音符の音程から、後ろの音符のピッチに滑らかにピッチを変化させながら歌う歌唱方法がある。ピッチやアンプリチュードが変化するのは当然であるが、さらに、前後２つの音符の発音が同じ（例えば同じ「あ」）だとしても、フォルマント周波数などの音声の周波数特性が微妙に変化する。

【００７３】実際にピッチを変化させて歌った音声の変化を始点から終点まで解析して求めたＮＮテンプレートを使うことによって、そのような音程の異なる音符の境界に、自然な音楽的表情を、与えることができる。

【００７４】実際の音楽における旋律では、２オクターブ２４音の音域としたとしても、ピッチ変化の組合せは非常に多い。しかし、実際にはピッチの絶対値が異なっていてもピッチ差が近いテンプレートで代用することができるので全ての組合せについてＮＮテンプレートを用意する必要はない。

【００７５】ＮＮテンプレートの選択においては、後述するように、ピッチの絶対値が近いものよりも、ピッチの変化幅が近いテンプレートを優先的に選択する。選択されたＮＮテンプレートは、後述するタイプ３の方法で適用する。

【００７６】このとき、ピッチの変化幅が近いＮＮテンプレートを優先的に選ぶのは、ピッチの大きく変動する部分から作成したＮＮテンプレートには大きな値が入っている可能性があり、それをピッチの変化幅が少ない部分に適用した場合には元のＮＮテンプレートの持っている変化の形状を保てなくなり、変化が不自然になる可能性があるからである。

【００７７】なお、ある特定の音素、例えば「あ」のピッチの変化している音声から求めたＮＮテンプレートを、全ての音素のピッチ変化に代用して使うことも可能であるが、データサイズが大きくても問題がない環境であれば、音素ごとに何パターンかピッチを変化させてＮＮテンプレートを用意するほうが、より単調でない豊かな合成音声が可能となる。

【００７８】次に、データベース４に記録されているテンプレートの適用方法を説明する。テンプレートの適用とは、入力データＳｃｏｒｅ上のある区間に対して、テンプレートの時間長を伸縮して、基準点となる１つ又は複数の特徴パラメータにテンプレートの特徴パラメータとの差分を加算して、Ｓｃｏｒｅのある区間と同じ時間長を持つ特徴パラメータ、ピッチの列を得ることである。具体的にはタイプ１からタイプ４までの４種類のテンプレートの適用方法がある。以下の説明ではテンプレートを｛Ｐ（ｔ），Ｐｉｔｃｈ（ｔ），Ｔ｝であらわす。

【００７９】まずタイプ１によるテンプレートの適用を説明する。タイプ1は、始点指定タイプによるテンプレートの適用方法である。入力データＳｃｏｒｅの長さＴの区間Ｋに対するタイプ１によるテンプレートの適用は、下記式（Ｄ）に従って時刻ｔでの特徴パラメータP'_tを求めることである。なおP_tは区間Ｋの時刻ｔの特徴パラメータである。

【００８０】

【数式６】

$$P'_t = P_t + P(t \cdot T / T') - P(0)$$

…（Ｄ）

なお、時刻ｔ＝０にテンプレート及び区間Ｋの始点があるとする。この式（Ｄ）はテンプレートの始点からの変化分を時刻ｔの特徴パラメータに加算することを意味する。

【００８１】タイプ１は、テンプレートを主にノートリ

リース部分の特徴パラメータに適用する場合に用いる。何故なら、ノートリリースの開始部分では、定常部分の音声が存在する為、ノートリリースの開始部分でパラメータの連続性、つまりは音声の連続性を保つ必要があり、ノートリリースの終端部は無音であるので、その必要がないからである。

【００８２】次にタイプ２によるテンプレートの適用方法を説明する。タイプ２は、終点指定タイプによるテンプレートの適用方法である。入力データＳｃｏｒｅの長さＴの区間Ｋに対するタイプ２によるテンプレートの適用は、下記式（Ｅ）に従って時刻ｔでの特徴パラメータＰ’$_t$を求めることである。なおＰ$_t$は区間Ｋの時刻ｔの特徴パラメータである。

【００８３】

【数式７】

$$P'_t = P_t + P(t \cdot T / T') - P(T)$$

…（Ｅ）

なお、時刻ｔ＝０にテンプレート及び区間Ｋの始点があるとする。この式（Ｅ）はテンプレートの終点からの変化分を時刻ｔの特徴パラメータに加算することを意味する。

【００８４】タイプ２は、テンプレートを主にノートアタック部分の特徴パラメータに適用する場合に用いる。何故なら、ノートアタックの後方部分では、定常部分の音声が存在する為、ノートアタックの後方部分でパラメータの連続性、つまりは音声の連続性を保つ必要があり、ノートアタックの開始部分は無音であるので、その必要がないからである。

【００８５】次にタイプ３によるテンプレートの適用方法を説明する。タイプ３は、両点指定タイプによるテンプレートの適用方法である。入力データＳｃｏｒｅの長さＴの区間Ｋに対するタイプ３によるテンプレートの適用は、下記式（Ｆ）に従って時刻ｔでの特徴パラメータＰ’$_t$を求めることである。なおＰ$_t$は区間Ｋの時刻ｔの特徴パラメータである。

【００８６】

【数式８】

$$P'_t = P_0 + \frac{t}{T'}(P_t - P_0) + \left(P(t \cdot T / T') - \frac{t}{T'}(P(T) - P(0))\right)$$

…（Ｆ）

なお、時刻ｔ＝０にテンプレート及び区間Ｋの始点があるとする。この式（Ｆ）はテンプレートの始点と終点を結んだ直線との差を、区間Ｋの始点と終点を結んだ直線に加算することを意味する。

【００８７】次にタイプ４によるテンプレートの適用方法を説明する。タイプ４は、ステーショナリータイプによるテンプレートの適用方法である。入力データＳｃｏｒｅの長さＴの区間Ｋに対するタイプ２によるテンプレートの適用は、下記式（Ｇ）に従って時刻ｔでの特徴パラメータＰ’$_t$を求めることである。なおＰ$_t$は区間Ｋの時刻ｔの特徴パラメータである。

【００８８】

【数式９】

$$P'_t = P_t + P(t \bmod T) - P(0)$$

…（Ｇ）

なお、時刻ｔ＝０にテンプレート及び区間Ｋの始点があるとする。この式（Ｇ）は区間Ｋに対してテンプレートの始点からの特徴パラメータの変化分を加算することをＴ毎に繰り返すことを意味する。

【００８９】タイプ４は、主にステーショナリー部分に適用する場合に用いる。このタイプ４は、比較的長時間の音声の定常的部分に自然な揺らぎを与える効果をもっている。

【００９０】図９は、特徴パラメータ発生処理を表すフローチャートである。この処理により、ある時刻ｔにおける特徴パラメータを発生させる。この特徴パラメータ発生処理を、ある一定時刻毎に時刻ｔを増加させながら、繰り返し行うことにより、フレーズ、曲といった単位の音声を合成することが出来る。

【００９１】ステップＳＡ１では、特徴パラメータ発生処理を開始して次のステップＳＡ２に進む。

【００９２】ステップＳＡ２では、入力データＳｃｏｒｅの時刻ｔにおける各トラックの値を取得する。具体的には、入力データＳｃｏｒｅ中の時刻ｔにおける音韻名、アーティキュレーション又はステーショナリーの区別、ノートアタック、ノートトゥノート又はノートリリースの区別、ピッチ、ダイナミクス値、及びオープニング値を取得する。その後次のステップＳＡ３に進む。

【００９３】ステップＳＡ３では、ステップＳＡ２で取得した入力データＳｃｏｒｅの各トラックの値に基づき、必要なテンプレートを音韻テンプレートデータベースＰＤＢとノートテンプレートデータベースＮＤＢから読み込む。その後次のステップＳＡ４に進む。

【００９４】このステップＳＡ３での音韻テンプレートの読み込みは、例えば、以下の手順で行われる。時刻ｔでの音韻がアーティキュレーションであると判断すると、アーティキュレーションテンプレートデータベースを検索して、先頭と後続の音韻名が一致して、かつピッチが一番近いテンプレートを読み込む。

【００９５】一方、時刻ｔでの音韻がステーショナリーであると判断すると、ステーショナリーテンプレートデータベースを検索して、音韻名が一致して、かつピッチが一番近いステーショナリーテンプレートを読み込む。

【００９６】また、ノートテンプレートの読み込みは、以下のように行われる。例えば、時刻ｔのノートトラックがノートアタックであると判断した場合は、ＮＡテンプレートデータベースＮＡＤＢを検索して、音韻名が一

致して、かつピッチが一番近いテンプレートを読み込む。

【００９７】また、例えば、時刻ｔのノートトラックがノートリリースであると判断した場合は、ＮＲテンプレートデータベースＮＲＤＢを検索して、音韻名が一致して、かつピッチが一番近いテンプレートを読み込む。

【００９８】さらに、例えば、時刻ｔのノートトラックがノートトゥノートであると判断した場合は、ＮＮテンプレートデータベースＮＮＤＢを検索して、音韻名が一致して、かつ始点ピッチと終了時刻ピッチを元に以下の式（Ｈ）で求められる距離ｄが一番近くなるテンプレートを読み込む。以下の式（Ｈ）は、周波数の変化量と平均値を重み付けして加算した値を元に距離尺度としている。

【００９９】

【数式１０】

$$d = 0.8 \cdot |TempInterval - Interval| + 0.2 \cdot |TempAve - Ave| \quad \cdots (H)$$

【数式１１】

ここで、

$$TempInterval = |\text{テンプレートの始点ピッチ} - \text{テンプレートの終点ピッチ}|$$

$$TempAve = (\text{テンプレートの始点ピッチ} + \text{テンプレートの終点ピッチ})/2$$

$$Interval = |\text{ノートトラック上の始点ピッチ} - \text{ノートトラック上の終点ピッチ}|$$

$$Ave = (\text{ノートトラック上の始点ピッチ} + \text{ノートトラック上の終点ピッチ})/2$$

上記式（Ｈ）で求めた距離ｄに基づき、テンプレートを読み込むことにより、ピッチの絶対値が近いものよりも、ピッチの変化幅が近いテンプレートを優先的に選択するようにしている。

【０１００】ステップＳＡ４では、ノートトラックの現在時刻ｔと同じ属性を持つ領域の開始時刻及び終了時刻を求め、音韻トラックがステーショナリーである場合はノートアタック、ノートトゥノート又はノートリリースの区別にしたがって、開始時刻あるいは終了時刻又は双方の特徴パラメータを取得若しくは算出する。その後次のステップＳＡ５に進む。

【０１０１】時刻ｔのノートトラックがノートアタックである場合には、Ｔｉｍｂｒｅデータベース ＴＤＢを検索して、音韻名及びノートアタック終了時刻のピッチが一致する特徴パラメータを読み込む。

【０１０２】ピッチが一致する特徴パラメータがないときには、音韻名が一致し、かつノートアタック終了時刻のピッチをはさむ２つの特徴パラメータを取得して、これらを補間することによりノートアタック終了時刻の特徴パラメータを算出する。補間方法の詳細は後述する。

【０１０３】時刻ｔのノートトラックがノートリリースである場合には、Ｔｉｍｂｒｅデータベース ＴＤＢを検索して、音韻名及びノートアタック開始時刻のピッチが一致する特徴パラメータを読み込む。

【０１０４】ピッチが一致する特徴パラメータがないときには、音韻名が一致し、かつノートリリース開始時刻のピッチをはさむ２つの特徴パラメータを取得して、これらを補間することによりノートリリース開始時刻の特徴パラメータを算出する。補間方法の詳細は後述する。

【０１０５】時刻ｔのノートトラックがノートトゥノートである場合には、Ｔｉｍｂｒｅデータベース ＴＤＢを検索して、音韻名とノートトゥノート開始時刻のピッチが一致する特徴パラメータ及び音韻名とノートトゥノート終了時刻が一致する特徴パラメータを読み込む。

【０１０６】ピッチが一致する特徴パラメータがないときには、音韻名が一致し、かつノートトゥノート開始（終了）時刻のピッチをはさむ２つの特徴パラメータを取得して、これらを補間することによりノートトゥノート開始（終了）時刻の特徴パラメータを算出する。補間方法の詳細は後述する。

【０１０７】なお、音韻トラックがアーティキュレーションである場合は開始時刻及び終了時刻の特徴パラメータを取得若しくは算出する。この場合は、Ｔｉｍｂｒｅデータベース ＴＤＢを検索して、音韻名とアーティキュレーション開始時刻のピッチが一致する特徴パラメータ及び音韻名とアーティキュレーション終了時刻が一致する特徴パラメータを読み込む。

【０１０８】ピッチが一致する特徴パラメータがないときには、音韻名が一致し、かつアーティキュレーション開始（終了）時刻のピッチをはさむ２つの特徴パラメータを取得して、これらを補間することによりアーティキュレーション開始（終了）時刻の特徴パラメータを算出する。

【０１０９】ステップＳＡ５では、ステップＳＡ４で求めた始点、終了時刻の特徴パラメータとピッチに対して、ステップＳＡ３で読み込んだテンプレートを適用して、時刻ｔにおけるピッチとダイナミクスを求める。

【０１１０】時刻ｔのノートトラックがノートアタックならば、ノートアタック部分に対してステップＳＡ４で求めたノートアタック部分の終了時刻の特徴パラメータを使いタイプ２でＮＡテンプレートを適用する。テンプレートを適用した後の時刻ｔにおけるピッチとダイナミクス（ＥＧａｉｎ）を記憶する。

【０１１１】一方、時刻ｔのノートトラックがノートリ

リースならば、ノートリリース部分に対してステップＳＡ４で求めたノートリリース始点の特徴パラメータを使いタイプ１でＮＲテンプレートを適用する。テンプレートを適用した後の時刻ｔにおけるピッチとダイナミクス（ＥＧａｉｎ）を記憶する。

【０１１２】また、時刻ｔのノートトラックがノートトゥノートならば、ノートトゥノート部分に対してステップＳＡ４で求めたノートトゥノートの始点及び終了時刻における特徴パラメータを使い、その区間に対してタイプ３でＮＮテンプレートを適用する。テンプレートを適用した後の時刻ｔにおけるピッチとダイナミクス（ＥＧａｉｎ）を記憶する。

【０１１３】さらに、時刻ｔのノートトラックが上記のいずれでもない場合には、入力データＳｃｏｒｅのピッチとダイナミクス（ＥＧａｉｎ）を記憶する。

【０１１４】以上のいずれかの処理を行ったら、次のステップＳＡ６に進む。

【０１１５】ステップＳＡ６では、ステップＳＡ２で求めた各トラックの値から、時刻ｔの音韻がアーティキュレーションであるか否かを判断する。アーティキュレーションである場合には、ＹＥＳの矢印で示すステップＳＡ９に進む。アーティキュレーションでない場合、すなわち時刻ｔの音韻がステーショナリーである場合には、ＮＯの矢印で示すステップＳＡ７に進む。

【０１１６】ステップＳＡ７では、ステップＳＡ２で求めた時刻ｔにおける音韻名と、ステップＳＡ５で求めたピッチ、ダイナミクスをインデックスとして、Ｔｉｍｂｒｅデータベース ＴＤＢから特徴パラメータを読み込み補間する。読み込みと補間の方法は、ステップＳＡ４で行ったものと同様である。その後、ステップＳＡ８に進む。

【０１１７】ステップＳＡ８では、ステップＳＡ７で求めた時刻ｔにおける特徴パラメータ及びピッチに対して、ステップＳＡ３で求めたステーショナリーテンプレートをタイプ４で適用する。

【０１１８】このステップＳＡ８で、ステーショナリーテンプレートを適用することで、時刻ｔでの特徴パラメータ及びピッチが更新され、ステーショナリーテンプレートの持つ音声の揺らぎが加えられる。その後、ステップＳＡ１０に進む。

【０１１９】ステップＳＡ９では、ステップＳＡ４で求めたアーティキュレーション部分の開始時刻及び終了時刻の特徴パラメータに、ステップＳＡ３で読み込んだアーティキュレーションテンプレートを適用して、時刻ｔでの特徴パラメータ及びピッチを求める。その後、ステップＳＡ１０に進む。

【０１２０】ただし、テンプレートの適用方法は有声音（Ｖ）から無声音（Ｕ）への変化の場合はタイプ１で行い、無声音（Ｕ）から有声音（Ｖ）への変化の場合はタイプ２で行い、有声音（Ｖ）から有声音（Ｖ）又は無声音（Ｕ）からから無声音（Ｕ）への変化の場合はタイプ３で行う。

【０１２１】上記のようにテンプレートの適用方法を変えるのは、有声部分での連続性を保ちつつ、テンプレートに含まれている自然な音声の変化を再現する為である。

【０１２２】ステップＳＡ１０では、ステップＳＡ８若しくはステップＳＡ９で求められた特徴パラメータに対して、ＮＡテンプレート、ＮＲテンプレート、ＮＮテンプレートのいずれかを適用する。ただし、ここでは、特徴パラメータのＥＧａｉｎに対しては、テンプレートを適用しない。その後次のステップＳＡ１１に進み、特徴パラメータ発生処理を終了する。

【０１２３】このステップＳＡ１０でのテンプレートの適用は、時刻ｔでのノートトラックがノートアタックである場合には、ステップＳＡ３で求めた、ＮＡテンプレートをタイプ２により適用して、特徴パラメータを更新する。

【０１２４】時刻ｔでのノートトラックがノートリリースである場合には、ステップＳＡ３で求めた、ＮＲテンプレートをタイプ１により適用して、特徴パラメータを更新する。

【０１２５】時刻ｔでのノートトラックがノートトゥノートである場合には、ステップＳＡ３で求めた、ＮＮテンプレートをタイプ３により適用して、特徴パラメータを更新する。

【０１２６】ただし上記いずれの場合にも、ここでは、特徴パラメータのＥＧａｉｎに対しては、テンプレートを適用しない。また、ピッチについても、このステップＳＡ１０の前のステップで求められたものをそのまま使用する。

【０１２７】以下に、図９のステップＳＡ４で行う特徴パラメータの補間について説明する。特徴パラメータの補間には、２つの特徴パラメータの補間と、１つの特徴パラメータからの推定がある。

【０１２８】人間が音声を発声するときにピッチを変化させると声帯波形（肺からの空気と声帯の振動によって発生する音源波形）が変化することが知られており、またフォルマントもピッチによって変化することが知られている。ある特定のピッチで歌った音声から得られた特徴パラメータを他のピッチの音声を合成するときにそのまま流用した場合には、ピッチを変えても同じような声の音色になってしまい不自然になってしまう。

【０１２９】それを避けるために人間の歌唱音域である２～３オクターブの音域中、対数軸で、ほぼ等間隔で３点程度のピッチを選び、特徴パラメータをＴｉｍｂｒｅデータベース ＴＤＢに保存しておく。Ｔｉｍｂｒｅデータベース ＴＤＢ中にあるピッチ以外のピッチの音声を合成する場合には、２つの特徴パラメータの補間（直線補間）若しくは１つの特徴パラメータからの推定（外挿）

によって特徴パラメータが求められる。

【０１３０】この方法によって、ピッチが変化したときの音声の特徴パラメータの変化を擬似的に表現することができる。また、ピッチの異なる特徴パラメータを３点程度持つのは、同じ音素、同じピッチの発生でもそのときによって特徴パラメータには変動があり、３点程度から補間して求めた場合とさらに細かく分割して求めた場合との差は余り意味がないからである。

【０１３１】２つの特徴パラメータの補間は、例えば、２つの特徴パラメータとそれぞれのピッチの組｛Ｐ１，ｆ１［ｃｅｎｔｓ］｝、｛Ｐ２，ｆ２［ｃｅｎｔｓ］｝が与えられたときに、時刻ｔのピッチｆ１［ｃｅｎｔｓ］における特徴パラメータを、以下の式（Ｉ）により直線補間して求めることにより行われる。

【０１３２】

【数式１２】

$$P=\frac{|f_2-f|}{|f_1-f|+|f_2-f|}P_1+\frac{|f_1-f|}{|f_1-f|+|f_2-f|}P_2=P_1+(P_2+P_1)\frac{f-f_1}{|f_2-f_1|}$$

…（Ｉ）

上記式（Ｉ）では、データベースのインデックスがピッチ1個だけの場合を考えたが、一般的にインデックスがＮ個ある場合でも、目標を囲む近傍のＮ＋1個のデータをもとに、以下の式（Ｉ'）を用いて、目標のインデックスｆの代理として使用する特徴パラメータを補間して求めることが出来る。なお、P_iは、近傍のｉ番目の特徴パラメータであり、f_iはそのインデックスである。

【０１３３】

【数式１３】

$$P=\sum_{i=1}^{N+1}\frac{\left(\sum_{j=1}^{N+1}|fj-f|\right)-fi}{\sum_{j=1}^{N+1}|fj-f|}Pi$$

…（Ｉ'）

１つの特徴パラメータからの推定は、データベースに含まれるデータの音域を外れる音声の特徴パラメータを推定するときに用いる。

【０１３４】これは、データベースの音域よりもピッチの高い音声を合成する場合に、データベース中の最もピッチの高い特徴パラメータをそのまま利用すると、明らかに音質が劣化するからである。

【０１３５】また、データベースの音域よりもピッチの低い音声を合成する場合に、最もピッチの低い特徴パラメータを利用すると同様に音質が劣化するからである。そこで本実施例では実際の音声データの観察からの知見に基づいた規則を使って、以下のように特徴パラメータを変化させて劣化を防いでいる。

【０１３６】まず、データベースの音域よりも高いピッチ（目標ピッチ）の音声を合成する場合を説明する。

【０１３７】まず、目標ピッチＴａｒｇｅｔＰｉｔｃｈ［ｃｅｎｔｓ］からデータベース中の最も高いピッチＨｉｇｈｅｓｔＰｉｔｃｈ［ｃｅｎｔｓ］を引いた値ＰｉｔｃｈＤｉｆｆ［ｃｅｎｔｓ］を求める。

【０１３８】次に、データベースから最も高いピッチを持つ特徴パラメータを読み出して、その内の励起レゾナンス周波数ＥｐＲＦｒｅｑ及び第ｉフォルマント周波数ＦｏｒｍａｎｔＦｒｅｑ$_i$に、それぞれ上記ＰｉｔｃｈＤｉｆｆ［ｃｅｎｔｓ］を加算して、ＥｐＲＦｒｅｑ'、ＦｏｒｍａｎｔＦｒｅｑ$_i$'に置き換えたものを目標ピッチの特徴パラメータとして使う。

【０１３９】次に、データベースの音域よりも低いピッチ（目標ピッチ）の音声を合成する場合を説明する。

【０１４０】まず、目標ピッチＴａｒｇｅｔＰｉｔｃｈ［ｃｅｎｔｓ］からデータベース中の最も低いピッチＬｏｗｅｓｔＰｉｔｃｈ［ｃｅｎｔｓ］を引いた値ＰｉｔｃｈＤｉｆｆ［ｃｅｎｔｓ］を求める。

【０１４１】次に、データベースから最も低いピッチを持つ特徴パラメータを読み出して、以下のようにパラメータを置き換えて目標ピッチの特徴パラメータとして用いる。

【０１４２】まず、励起レゾナンス周波数ＥｐＲＦｒｅｑ及び第１から第４フォルマント周波数ＦｏｒｍａｎｔＦｒｅｑ（１≦ｉ≦４）を、それぞれ下記式（Ｊ１）及び（Ｊ２）を用いて、ＥｐＲＦｒｅｑ'、ＦｏｒｍａｎｔＦｒｅｑ$_i$'に置き換える。

【数式１４】

$$ERFreq'=ERFreq+0.25\times PitchDiff$$

【数式１５】

$$FormantFreq_i'=FormantFreq_i+0.25\times PitchDiff$$

さらに、ピッチが低くなるほどバンド幅が狭くなるように、励起レゾナンスバンド幅ＥＲＢＷ及び第１から第３フォルマントのバンド幅ＦｏｒｍａｎｔＢＷ$_i$（１≦ｉ≦３）をそれぞれ下記式（Ｊ３）、（Ｊ４）のＥＲＢＷ'、ＦｏｒｍａｎｔＢＷ$_i$'に置き換える。

【０１４３】

【数式１６】

$$ERBW'=\frac{ERBW}{1-3\times PichDiff/1200}$$

…（Ｊ３）

【数式１７】

$$FormantFreq_i'=FormantFreq_i+0.25\times PitchDiff$$

…（Ｊ４）

さらに、第１から第４フォルマントのアンプリチュード

ＦｏｒｍａｎｔＡｍｐ１～ＦｏｒｍａｎｔＡｍｐ４を下記式（Ｊ５）～（Ｊ８）に従いＰｉｔｃｈＤｉｆｆに比例させて大きくして、ＦｏｒｍａｎｔＡｍｐ１’～ＦｏｒｍａｎｔＡｍｐ４’に置き換える。

【０１４４】

【数式１８】

$$FormantAmp_1' = FormantAmp_1 - 8 \times PitchDiff / 1200$$

…（Ｊ５）

【数式１９】

$$FormantAmp_2' = FormantAmp_2 - 5 \times PitchDiff / 1200$$

…（Ｊ６）

【数式２０】

$$FormantAmp_3' = FormantAmp_3 - 12 \times PitchDiff / 1200$$

…（Ｊ７）

【数式２１】

$$FormantAmp_4' = FormantAmp_4 - 15 \times PitchDiff / 1200$$

…（Ｊ８）

さらに、スペクトル・エンベロープの傾きＥｓｌｏｐｅを下記式（Ｊ９）に従いＥｓｌｏｐｅ’に置き換える。

【０１４５】

【数式２２】

$$ESlope' = ESlope \times (1 - 4 \times PitchDiff / 1200)$$

…（Ｊ９）

図４に示すような、ピッチ、ダイナミクス、オープニングをインデックスとしてＴｉｍｂｒｅデータベースＴＤＢを作成することが好ましいが、時間的、データベースサイズ的な制約がある場合には、本実施例のように、図３に示すような、ピッチのみをインデックスとしたデータベースを用いることになる。

【０１４６】そのような場合に、ダイナミクス関数や、オープニング関数を用いて、ピッチのみをインデックスとした特徴パラメータを変化させ、あたかも、ピッチ、ダイナミクス、オープニングをインデックスとして作成したＴｉｍｂｒｅデータベースＴＤＢを使用したかのような効果を擬似的に得る事が出来る。

【０１４７】すなわち、ピッチのみを変化させて録音した音声を使用して、ピッチ、ダイナミクス、オープニングを変化させて録音した音声を使用したかのような効果を得る事が出来る。

【０１４８】ダイナミクス関数及び、オープニング関数は、ダイナミクス、オープニングを変化させて発声した実際の音声と、特徴パラメータの相関関係を分析して得る事が出来る。以下に、ダイナミクス関数及び、オープニング関数の例をあげ、その適用方法を説明する。

【０１４９】図１０は、ダイナミクス関数の一例を表すグラフである。図１０（Ａ）は、関数ｆＥＧを表すグラフであり、図１０（Ｂ）は、関数ｆＥＳを表すグラフであり、図１０（Ｃ）は、関数ｆＥＳＤを表すグラフである。

【０１５０】これらの、図１０（Ａ）～（Ｃ）に示される関数ｆＥＧ、ｆＥＳ、ｆＥＳＤを利用して、ダイナミクス値を特徴パラメータＥｘｃｉｔａｔｉｏｎＧａｉｎ（ＥＧ）、ＥｘｃｉｔａｔｉｏｎＳｌｏｐｅ（ＥＳ）、ＥｘｃｉｔａｔｉｏｎＳｌｏｐｅＤｅｐｔｈ（ＥＳＤ）に反映させる。

【０１５１】図１０（Ａ）～（Ｃ）の関数ｆＥＧ、ｆＥＳ、ｆＥＳＤの入力は、全てダイナミクス値であり、０から１までの値をとる。このダイナミクス値をｄｙｎとして、関数ｆＥＧ、ｆＥＳ、ｆＥＳＤを使い、下記式（Ｋ１）～（Ｋ３）で、特徴パラメータＥＧ’、ＥＳ’、ＥＳＤ’を求め、ダイナミクス値（ｄｙｎ）の時の特徴パラメータとして用いる。

【０１５２】

【数式２３】

$$EG' = fEG(dyn)$$

…（Ｋ１）

【数式２４】

$$ES' = ES \times fES(dyn)$$

…（Ｋ２）

【数式２５】

$$ESD' = ESD + fESD(dyn)$$

…（Ｋ３）

なお、図１０（Ａ）～（Ｃ）の関数ｆＥＧ、ｆＥＳ、ｆＥＳＤは、一例であり、歌唱者によって様々な関数を用意することにより、より自然性を持った音声合成を行うことが出来る。

【０１５３】図１１は、オープニング関数の一例を表すグラフである。図中、横軸は周波数（Ｈｚ）であり、縦軸はアンプリチュード（ｄＢ）である。

【０１５４】このオープニング関数をｆＯｐｅｎ（ｆｒｅｑ）とし、オープニング値をＯｐｅｎとして、以下の式（Ｌ１）により、励起レゾナンス周波数ＥＲＦｒｅｑ’を励起レゾナンス周波数ＥＲＦｒｅｑから求め、オープニング値（Ｏｐｅｎ）のときの特徴パラメータとして用いる。

【０１５５】

【数式２６】

$$ERFreq' = ERFreq + fOpen(ERFreq) \times (1 - Open)$$

…（Ｌ１）

また、以下の式（Ｌ２）により、ｉ番目のフォルマント周波数ＦｏｒｍａｎｔＦｒｅｑ$_i$’をｉ番目のフォルマント周波数ＦｏｒｍａｎｔＦｒｅｑ$_i$から求め、オープニング値（Ｏｐｅｎ）のときの特徴パラメータとして用いる。

【０１５６】

【数式２７】

$$FormantFreq_i' = FormantFreq_i + fOpen(FormantFreq_i) \times (1 - Open)$$

…（Ｌ２）

これにより、周波数０～５００Ｈｚにあるフォルマントのアンプリチュードをオープニング値に比例させて増減させることができ、合成音声に、唇開度による音声の変化を与えることが出来る。

【０１５７】なお、オープニング値を入力とする関数を歌唱者別に用意して、変化させることにより、合成音声をより多様化させることが出来る。

【０１５８】図１２は、本実施例によるテンプレートの第１の適用例を表す図である。図中（ａ）の楽譜による歌唱を本実施例により合成する場合を説明する。

【０１５９】この楽譜は、最初の２分音符の音程は「ソ」であり、強さは「ピアノ（弱く）」で「あ」という発音である。２つ目の２分音符の音程は「ド」であり、強さは「メゾフォルテ（やや強く）」で「あ」という発音である。２つの２分音符は、レガートで接続されているので、音と音の間に切れ目がなく滑らかに接続する。

【０１６０】ここで、「ソ」から「ド」への変化の時間は、入力データ（楽譜）とともに与えられるものとする。

【０１６１】まず、音符の音名から２つのピッチの周波数が得られる。その後、２つのピッチの終点と始点を直線で結んで、図中（ｂ）に示すように音符の境界部分のピッチを得ることが出来る。

【０１６２】次にダイナミクスであるが、これは、「ピアノ（弱く）」や「メゾフォルテ（やや強く）」といった強弱記号に対応した値をテーブルとして記憶しておき、これを使って数値に変換して２つの音符に対応するダイナミクス値を得る。このようにして得た２つのダイナミクス値を直線で結ぶことにより、図中（ｂ）に示すように音符の境界部分のダイナミクス値を得ることが出来る。

【０１６３】このようにして得て、ピッチと、ダイナミクス値をそのまま用いると、ピッチ、ダイナミクスが音符の境界部分で急激に変化してしまうので、レガートに接続する為、この音符の境界部分に、図中（ｂ）に示すようにＮＮテンプレートを適用する。

【０１６４】ここでは、ピッチとダイナミクスにだけ、ＮＮテンプレートを適用して、図中（ｃ）に示すような音符の境界部分が滑らかに接続されたピッチとダイナミクスを得る。

【０１６５】次に、図中（ｃ）に示す決定されたピッチとダイナミクス及び「あ」という音韻名をインデックスとして、Ｔｉｍｂｒｅデータベース ＴＤＢから、図中（ｄ）に示すような各時刻の特徴パラメータを求める。

【０１６６】ここで求めた各時刻の特徴パラメータに対して、図中（ｃ）に示す音韻名「あ」に対応するステーショナリーテンプレートを適用し、音符境界の接続部分以外の定常部分に音声の揺らぎを付加して、図中（ｅ）に示すような特徴パラメータを得る。

【０１６７】次に、図中（ｂ）でピッチとダイナミクスのみ適用したＮＮテンプレートの残り（フォルマント周波数など）を、図中（ｅ）に示す特徴パラメータに適用し、音符の境界部分のフォルマント周波数などに揺らぎを与えた図中（ｆ）で示す特徴パラメータを得る。

【０１６８】最後に、図中（ｃ）のピッチ、ダイナミクスと、図中（ｆ）の特徴パラメータを用いて、音声合成を行うことにより、図中（ａ）の楽譜で表す歌唱を合成することが出来る。

【０１６９】なお、図１２の（ｂ）で、ＮＮテンプレートを適用する部分の時間幅は、例えば、図１３に示すように長くすることが出来る。図１３に示すように、ＮＮテンプレートを適用する部分の時間幅を長くすると、ＮＮテンプレートが伸長されて適用されるので、ゆっくりとした変化を持つ歌唱音声を合成することが出来る。

【０１７０】また、逆に、ＮＮテンプレートを適用する時間幅を狭くすれば、早く滑らかに変化する歌唱音声を合成することが出来る。このようにＮＮテンプレートの適用時間を制御することで、変化のスピードをコントロールすることが出来る。

【０１７１】また、同じ時間で、ピッチをある高さから別の高さに変化させる場合でも、前半で急激に変化させ、後半はゆっくり変化させる歌い方があり、その逆もある。このように、ピッチの変化の道筋は何通りもあり、その違いは結果的に音楽的な聞こえ方の違いとなって現れる。そこで、このようなレガートの歌い方を変えて歌唱した音声から複数種類のＮＮテンプレートを作成して記録しておけば、様々なバリエーションを合成音声に持たせることが出来る。

【０１７２】さらに、音程（ピッチ）の変化の仕方には、上記のレガート奏法以外にも様々なものがあり、それらについても別にテンプレートを作成して記録するようにしてもよい。

【０１７３】例えば、レガートのように完全に連続的にピッチを変化させるのではなく、半音ごとにピッチを変化させたり、楽曲の長で使われる音階（例えば、ハ長調では、ドレミファソラシド）だけで飛び飛びに変化させたりする、いわゆるグリッサンド奏法がある。

【０１７４】この場合には、グリッサンドで実際に歌唱した音声から、ＮＮテンプレートを作成し、そのテンプレートを適用して２つの音符を滑らかに接続した歌唱を合成することが出来る。

【０１７５】なお、本実施例では、ＮＮテンプレート

は、同じ音韻でピッチが変化している場合だけを作成して記録しているが、例えば、「あ」から「え」のように違う音韻でピッチが変化している場合についても作成することができる。この場合は、NNテンプレートの数が多くなってしまうが、実際の歌唱により近づけることが出来る。

【０１７６】図１４は、本実施例によるテンプレートの第２の適用例を表す図である。図中（ａ）の楽譜による歌唱を本実施例により合成する場合を説明する。

【０１７７】この楽譜は、最初の２分音符の音程は「ソ」であり、強さは「ピアノ（弱く）」で「あ」という発音である。２つ目の２分音符の音程は「ド」であり、強さは「メゾフォルテ（やや強く）」で「え」という発音である。

【０１７８】ここで、「あ」から「え」へのアーティキュレーションの時間は、２つの音素の組合せ毎に固定値として設定しておくか、又は入力データとともに与えられるものとする。

【０１７９】まず、音符の音名から２つのピッチの周波数が得られる。その後、２つのピッチの終点と始点を直線で結んで、図中（ｂ）に示すように音符の境界部分（アーティキュレーション部分）のピッチを得ることが出来る。

【０１８０】次にダイナミクスであるが、これは、「ピアノ（弱く）」や「メゾフォルテ（やや強く）」といった強弱記号に対応した値をテーブルとして記憶しておき、これを使って数値に変換して２つの音符に対応するダイナミクス値を得る。このようにして得た２つのダイナミクス値を直線で結ぶことにより、図中（ｂ）に示すように音符の境界部分のダイナミクス値を得ることが出来る。

【０１８１】次に、図中（ｂ）に示す決定されたピッチとダイナミクス及び「あ」、「え」という音韻名をインデックスとして、ＴｉｍｂｒｅデータベースＴＤＢから、図中（ｃ）に示すような各時刻の特徴パラメータを求める。ただし、アーティキュレーション部分の特徴パラメータは、仮に音韻「あ」の終点部分と、音韻「え」の始点部分を直線補間した値である。

【０１８２】次に、図中（ｃ）に示すように、「あ」のステーショナリーテンプレート、「あ」から「え」へのアーティキュレーションテンプレート、「え」のステーショナリーテンプレートを先に求めた、特徴パラメータのそれぞれの該当部分に適用し、図中（ｄ）に示すような特徴パラメータを得る。

【０１８３】最後に、図中（ｂ）のピッチ、ダイナミクスと、（ｄ）の特徴パラメータを使って、音声合成を行う。

【０１８４】このようにすると、人間が実際に発声する場合と同様に、自然に「あ」から「え」に変化する歌唱音声を合成することが出来る。

【０１８５】なお、アーティキュレーションテンプレートも、NNテンプレートの場合と同様に、境界部分（アーティキュレーション部分）の長さを楽譜とともに与えられるようにしておけば、「あ」から「え」へのアーティキュレーションの時間を制御することができ、ゆっくりと変化する音声や、早く変化する音声を、１つのテンプレートを伸縮することで合成できる。すなわち、こうすることで、音韻の変化する時間を制御することが出来る。

【０１８６】図１５は、本実施例によるテンプレートの第３の適用例を表す図である。図中（ａ）の楽譜による歌唱を本実施例により合成する場合を説明する。

【０１８７】この楽譜は、音程が「ソ」で、発音は「あ」である全音符の強さを立ち上がりから次第に強くしていき、立下りで次第に弱くしていくものである。

【０１８８】この楽譜の場合は、ピッチ、ダイナミクスは図中（ｂ）に示すように平坦である。これらのピッチ、ダイナミクスの先頭にNAテンプレートを適用し、さらに音符の最後にNRテンプレートを適用して、図中（ｃ）で示すようなピッチとダイナミクスを求めて、決定する。

【０１８９】なお、NAテンプレート及びNRテンプレートを適用する長さは、クレッシェンド記号及びデクレッシェンド記号自身に長さを持たせて入力されているものとする。

【０１９０】次に、決定した図中（ｃ）のピッチ、ダイナミクス及び音韻名「あ」をインデックスとして、図中（ｄ）に示すようにアタックでもリリースでもない通常部分の特徴パラメータが求められる。

【０１９１】さらに、図中（ｄ）に示す通常部分の特徴パラメータに、ステーショナリーテンプレートを適用して、図中（ｅ）に示すような、揺らぎが与えられた特徴パラメータを求める。この（ｅ）の特徴パラメータを元に、アタック部分とリリース部分の特徴パラメータを求める。

【０１９２】アタック部分の特徴パラメータは、通常部分の始点（アタック部分の終点）に対して、音韻「あ」のNAテンプレートを前述のタイプ２の方法で適用して求める。

【０１９３】リリース部分の特徴パラメータは、通常部分の終点（リリース部分の始点）に対して、音韻「あ」のNRテンプレートを前述のタイプ１の方法で適用して求める。

【０１９４】このようにして、アタック部分、通常部分、リリース部分の特徴パラメータが、図中（ｆ）のように求められる。この特徴パラメータと、（ｃ）のピッチ、ダイナミクスを使用して、音声を合成することで、（ａ）の楽譜によるクレッシェンド、デクレッシェンドで歌った歌唱音声を得ることが出来る。

【０１９５】以上、本実施例に拠れば、実際の人間の歌

唱音声を分析して得られる音韻テンプレートを用いて、特徴パラメータに変動を与えるので、歌唱音声の持っている母音を長く伸ばした部分や、音韻が変化する部分の特徴を反映した自然な合成音声を生成することが出来る。

【０１９６】また、本実施例に拠れば、実際の人間の歌唱音声を分析して得られるノートテンプレートを用いて、特徴パラメータに変動を与えるので、単なる音量の違いだけでない、音楽的な強弱の表現力を持った合成音声を生成することが出来る。

【０１９７】さらに、本実施例に拠れば、ピッチ、ダイナミクス、オープニングなどの音楽表現度を細かく変化させたデータを用意しなくても、他に用意されているデータを補間して、用いることが出来るので、少ないサンプルですみ、データベースのサイズを小さくすることが出来るとともに、データベースの作成時間を短縮することが出来る。

【０１９８】さらに、また、本実施例に拠れば、音楽表現度として、ピッチのみをインデックスとしたデータベースを使用したとしても、オープニング及びダイナミクス関数を用いて、擬似的にピッチ、オープニング、ダイナミクスの３つの音楽表現度をインデックスとして持つデータベースを使用した場合に近い効果を得る事が出来る。

【０１９９】なお、本実施例では、図２に示したように、入力データＳｃｏｒｅとして、音韻トラックＰＨＴ、ノートトラックＮＴ、ピッチトラックＰＩＴ、ダイナミクストラックＤＹＴ、オープニングトラックＯＴを入力したが、入力データＳｃｏｒｅの構成はこれに限られない。

【０２００】例えば、図２の入力データＳｃｏｒｅに、ビブラートトラックを追加して用意してもよい。ビブラートトラックには、０～１のビブラート値が記録されている。

【０２０１】この場合、データベース４には、ビブラート値を引数として、ピッチ、ダイナミクスの時系列を返す関数、若しくはテーブルをビブラートテンプレートとして保存しておく。

【０２０２】そして、図４のステップＳＡ５のピッチ、ダイナミクスの計算において、このビブラートテンプレートを適用することで、ビブラート効果を与えたピッチ、ダイナミクスを得る事が出来る。

【０２０３】ビブラートテンプレートは、実際の人間の歌唱音声を分析することで得る事が出来る。

【０２０４】なお、本実施例は歌唱音声合成を中心に説明したが、歌唱音声に限られるものではなく、通常の会話の音声や楽器音なども同様に合成することができる。

【０２０５】なお、本実施例は、本実施例に対応するコンピュータプログラム等をインストールした市販のコンピュータ等によって、実施させるようにしてもよい。

【０２０６】その場合には、本実施例に対応するコンピュータプログラム等を、ＣＤ－ＲＯＭやフロッピー（登録商標）ディスク等の、コンピュータが読み込むことが出来る記憶媒体に記憶させた状態で、ユーザに提供してもよい。

【０２０７】そのコンピュータ等が、ＬＡＮ、インターネット、電話回線等の通信ネットワークに接続されている場合には、通信ネットワークを介して、コンピュータプログラムや各種データ等をコンピュータ等に提供してもよい。

【０２０８】以上実施例に沿って本発明を説明したが、本発明はこれらに制限されるものではない。例えば、種々の変更、改良、組合せ等が可能なことは当業者に自明であろう。

【０２０９】

【発明の効果】以上説明したように、本発明によれば、音質の劣化を最小限に抑えつつ、サイズを縮小した音声合成用データベースを提供することができる。

【０２１０】また、本発明によれば、よりリアルな人間の歌唱音声を合成して、違和感のない自然な状態で歌を歌わせることが可能な音声合成装置を提供することができる。

【図面の簡単な説明】

【図１】　本発明の実施例による音声合成装置１の構成を表すブロック図である。

【図２】　入力データＳｃｏｒｅの一例を示す概念図である。

【図３】　Ｔｉｍｂｒｅデータベース ＴＤＢの一例である。

【図４】　Ｔｉｍｂｒｅデータベース ＴＤＢの他の例である。

【図５】　ステーショナリーテンプレートデータベースの一例である。

【図６】　アーティキュレーションテンプレートデータベースの一例である。

【図７】　ＮＡテンプレートデータベース ＮＡＤＢの一例である。

【図８】　ＮＮテンプレートデータベース ＮＮＤＢの一例である。

【図９】　特徴パラメータ発生処理を表すフローチャートである。

【図１０】　ダイナミクス関数の一例を表すグラフである。

【図１１】　オープニング関数の一例を表すグラフである。

【図１２】　本実施例によるテンプレートの第１の適用例を表す図である。

【図１３】　本実施例によるテンプレートの第１の適用例の変形例を表す図である。

【図１４】　本実施例によるテンプレートの第２の適用

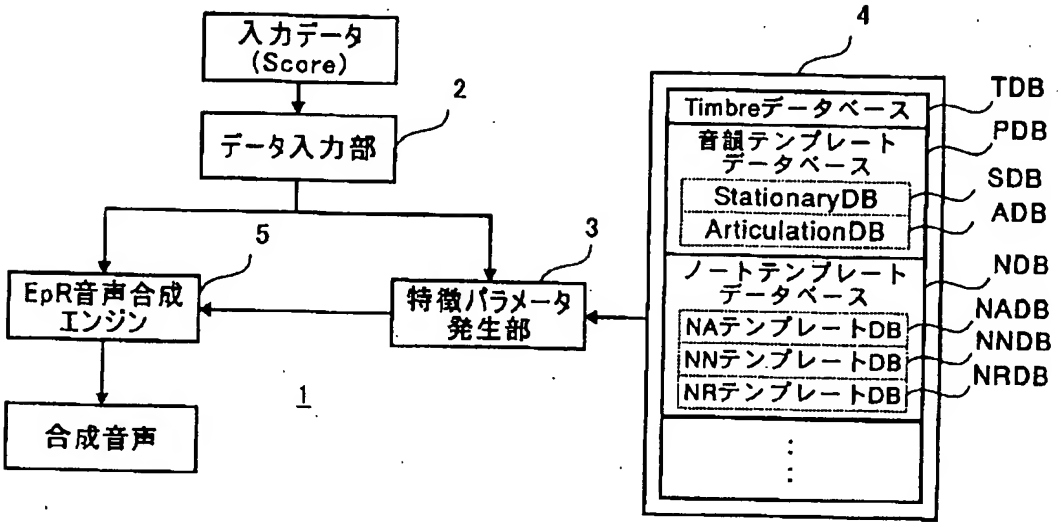
例を表す図である。

【図１５】　本実施例によるテンプレートの第３の適用例を表す図である。

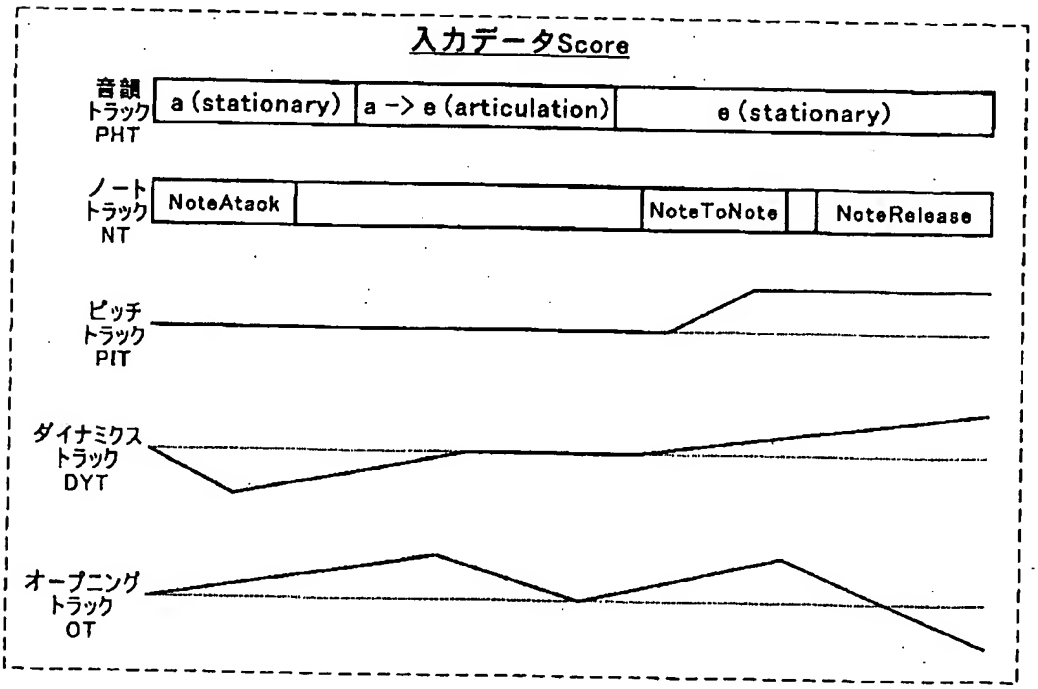
【符号の説明】

１…音声合成装置、２…データ入力部、３…特徴パラメータ発生部、４…データベース、５…ＥｐＲ音声合成エンジン

【図１】



【図２】



【図３】

音韻名	ピッチ[Hz]	特徴パラメータ
/a/	200	Pa1
/a/	300	Pa2
/a/	400	Pa3
/i/	200	Pi1
/i/	300	Pi2
:	:	:
/o/	500	Po4

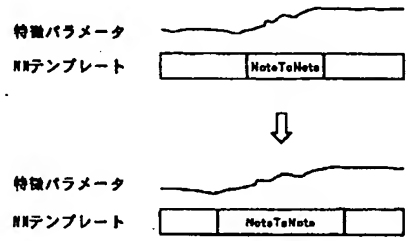
【図４】

音韻名	ピッチ[Hz]	ダイナミクス	オープニング	特徴パラメータ
/a/	200	0.8	0.4	Pa1
/a/	300	0.5	0.2	Pa2
/a/	400	0.6	0.8	Pa3
/i/	200	0.5	1	Pi1
/i/	300	0.3	0.7	Pi2
:	:	:	:	:
/o/	500	0.2	0.5	Po4

【図５】

音韻名	代表ピッチ[Hz]	特徴パラメータ
/a/	200	{Pa1(t),Pitch_a1(t),Ta1}
/a/	300	{Pa2(t),Pitch_a2(t),Ta2}
/a/	400	{Pa3(t),Pitch_a3(t),Ta3}
/i/	200	{Pi1(t),Pitch_i1(t),Ti1}
/i/	300	{Pi2(t),Pitch_i2(t),Ti2}
:	:	:
/o/	500	{Po4(t),Pitch_o4(t),To4}

【図１３】



【図６】

先頭音韻名	後続音韻名	代表ピッチ[Hz]	特徴パラメータ
/a/	/i/	200	{Pai1(t),Pitch_ai1(t),Tai1}
/a/	/i/	400	{Pai2(t),Pitch_ai2(t),Tai2}
/a/	/s/	300	{Pas1(t),Pitch_as1(t),Tas1}
/a/	/s/	500	{Pas2(t),Pitch_as2(t),Tas2}
:	:	:	:
/s/	/o/	500	{Pso3(t),Pitch_so3(t),Tso3}

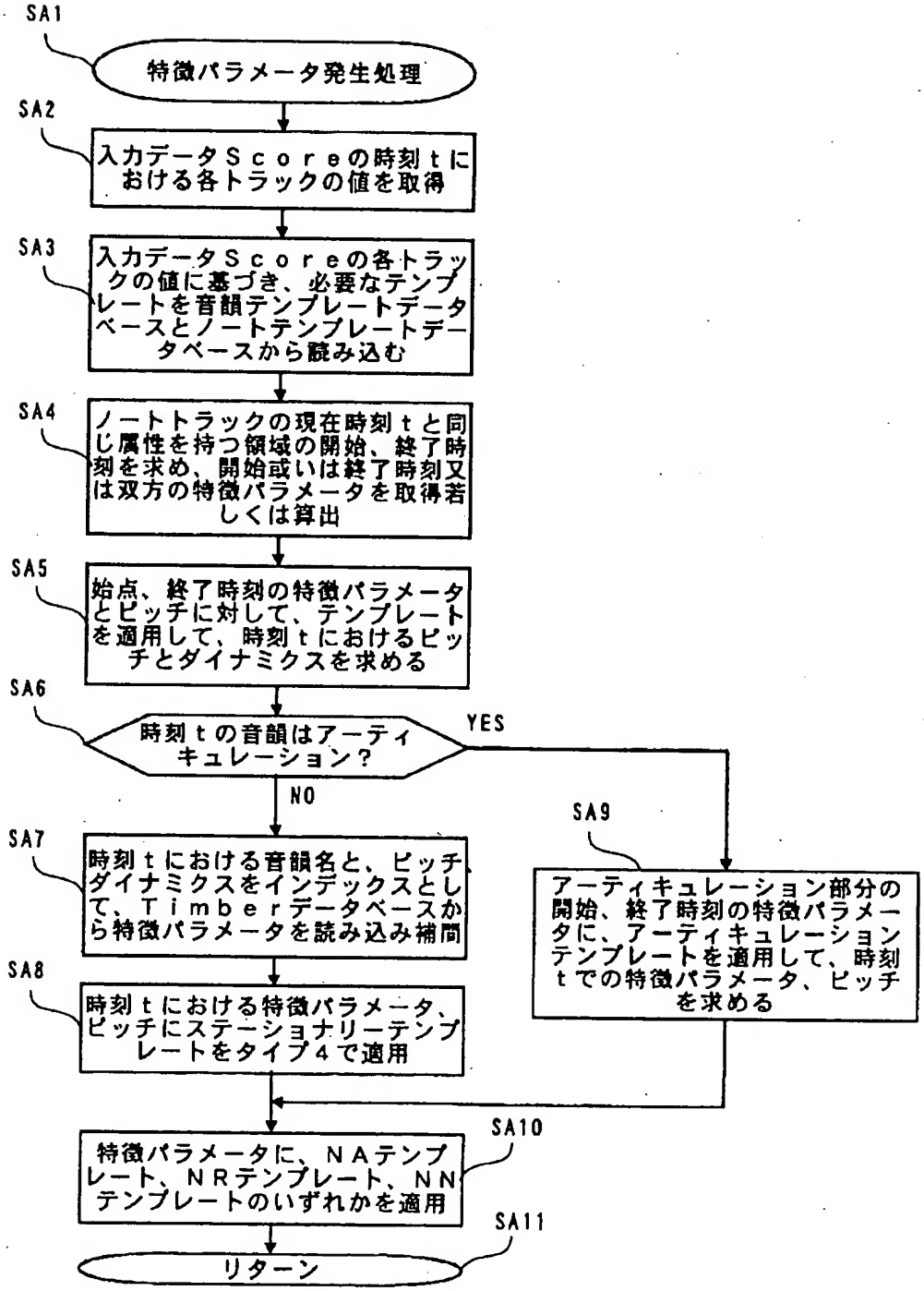
【図７】

音韻名	代表ピッチ[Hz]	特徴パラメータ
/a/	200	{PNA_a1(t),PitchNA_a1(t),TNA_a1}
/a/	300	{PNA_a2(t),PitchNA_a2(t),TNA_a2}
/a/	400	{PNA_a3(t),PitchNA_a3(t),TNA_a3}
/i/	200	{PNA_i1(t),PitchNA_i1(t),TNA_i1}
/i/	300	{PNA_i2(t),PitchNA_i2(t),TNA_i2}
:	:	:
/n/	500	{PNA_n4(t),PitchNA_n4(t),TNA_n4}

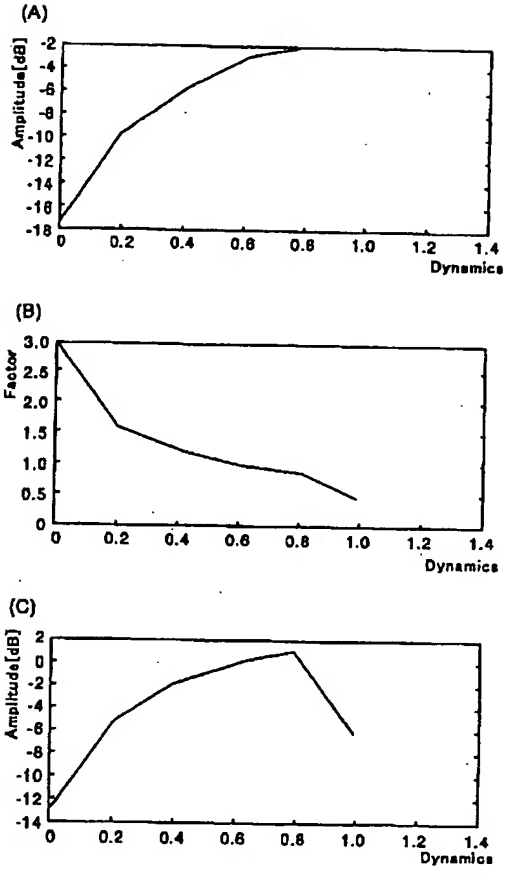
【図８】

音韻名	始点ピッチ[Hz]	終点ピッチ[Hz]	特徴パラメータ
/a/	200	100	{PNN_a1(t),PitchNN_a1(t),TNN_a1}
/a/	200	300	{PNN_a2(t),PitchNN_a2(t),TNN_a2}
/a/	400	300	{PNN_a3(t),PitchNN_a3(t),TNN_a3}
/i/	200	150	{PNN_i1(t),PitchNN_i1(t),TNN_i1}
/i/	300	200	{PNN_i2(t),PitchNN_i2(t),TNN_i2}
:	:	:	:
/n/	500	400	{PNN_n4(t),PitchNN_n4(t),TNN_n4}

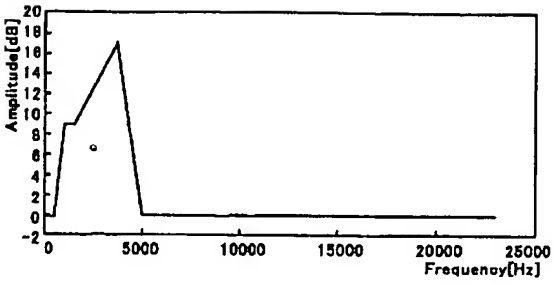
【図９】


SA1
特徴パラメータ発生処理
SA2
入力データＳｃｏｒｅの時刻ｔにおける各トラックの値を取得
SA3
入力データＳｃｏｒｅの各トラックの値に基づき、必要なテンプレートを音韻テンプレートデータベースとノートテンプレートデータベースから読み込む
SA4
ノートトラックの現在時刻ｔと同じ属性を持つ領域の開始、終了時刻を求め、開始或いは終了時刻又は双方の特徴パラメータを取得若しくは算出
SA5
始点、終了時刻の特徴パラメータとピッチに対して、テンプレートを適用して、時刻ｔにおけるピッチとダイナミクスを求める
SA6
時刻ｔの音韻はアーティキュレーション？
YES
NO
SA7
時刻ｔにおける音韻名と、ピッチ、ダイナミクスをインデックスとして、Ｔｉｍｂｅｒデータベースから特徴パラメータを読み込み補間
SA8
時刻ｔにおける特徴パラメータ、ピッチにステーショナリーテンプレートをタイプ４で適用
SA9
アーティキュレーション部分の開始、終了時刻の特徴パラメータに、アーティキュレーションテンプレートを適用して、時刻ｔでの特徴パラメータ、ピッチを求める
SA10
特徴パラメータに、ＮＡテンプレート、ＮＲテンプレート、ＮＮテンプレートのいずれかを適用
SA11
リターン

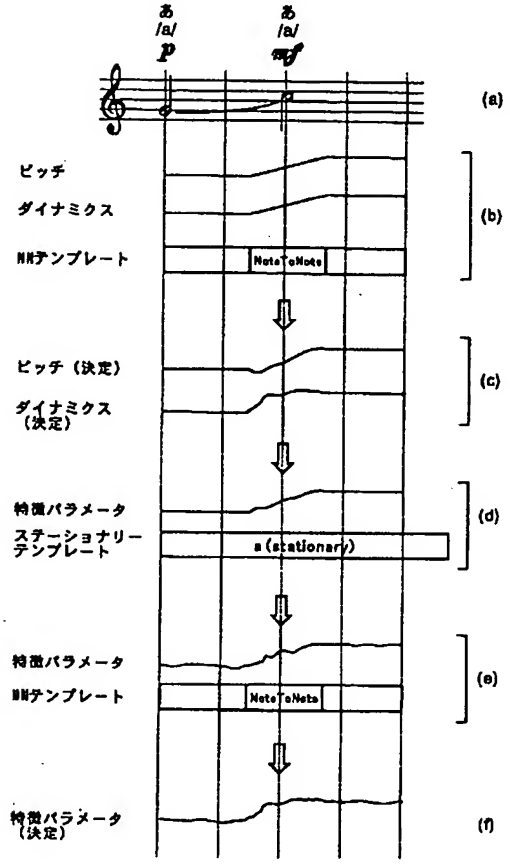
【図１０】


(A)
Amplitude[dB]
Dynamics
(B)
Factor
Dynamics
(C)
Amplitude[dB]
Dynamics

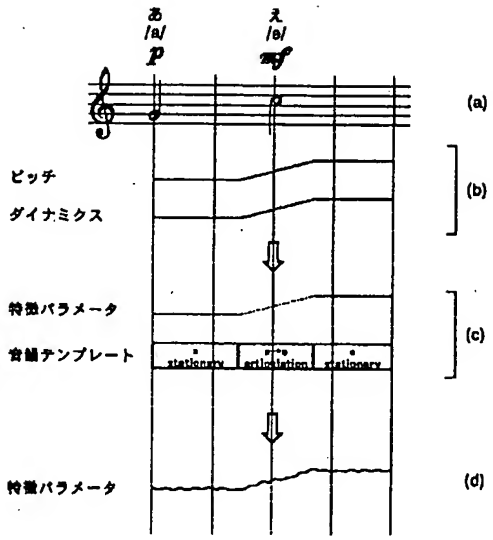
【図１１】


Amplitude[dB]
Frequency[Hz]

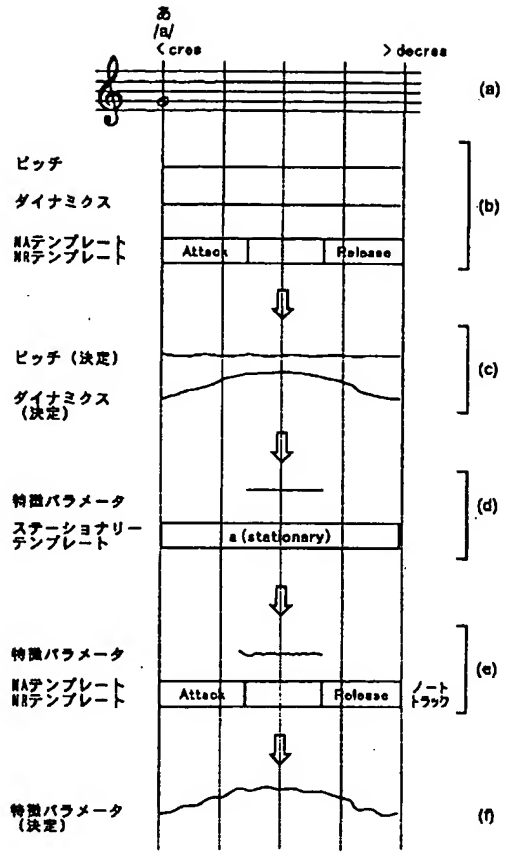
【図１２】


あ
/a/
あ
/a/
(a)
ピッチ
ダイナミクス
NNテンプレート
NoteToNote
(b)
ピッチ（決定）
ダイナミクス
（決定）
(c)
特徴パラメータ
ステーショナリー
テンプレート
a (stationary)
(d)
特徴パラメータ
NNテンプレート
NoteToNote
(e)
特徴パラメータ
（決定）
(f)

【図１４】


あ
/a/
p
え
/e/
mf
(a)
ピッチ
ダイナミクス
(b)
特徴パラメータ
音韻テンプレート
stationary
articulation
stationary
(c)
特徴パラメータ
(d)

【図１５】


あ
/a/
< cres
> decres
(a)
ピッチ
ダイナミクス
NAテンプレート
NRテンプレート
Attack
Release
(b)
ピッチ（決定）
ダイナミクス
（決定）
(c)
特徴パラメータ
ステーショナリー
テンプレート
a (stationary)
(d)
特徴パラメータ
NAテンプレート
NRテンプレート
Attack
Release
ノート
トラック
(e)
特徴パラメータ
（決定）
(f)